

APPEAL BRIEF UNDER 37 C.F.R. § 41.37

TABLE OF CONTENTS

| | <u>Page</u> |
|--|-------------|
| <u>1. REAL PARTY IN INTEREST</u> | 2 |
| <u>2. RELATED APPEALS AND INTERFERENCES</u> | 3 |
| <u>3. STATUS OF THE CLAIMS</u> | 4 |
| <u>4. STATUS OF AMENDMENTS</u> | 5 |
| <u>5. SUMMARY OF CLAIMED SUBJECT MATTER</u> | 6 |
| <u>6. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL</u> | 10 |
| <u>7. ARGUMENT</u> | 11 |
| <u>8. CLAIMS APPENDIX</u> | 18 |
| <u>9. EVIDENCE APPENDIX</u> | 32 |
| <u>10. RELATED PROCEEDINGS APPENDIX</u> | 33 |

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of: Wolfgang Kalthoff et al. Examiner: Thomas J. Dailey

Serial No.: 10/622,360

Group Art Unit: 2452

Filed: July 17, 2003

Docket: 2058.213US1

For: Collaborative design process

APPEAL BRIEF UNDER 37 CFR § 41.37

Mail Stop Appeal Brief- Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

The Appeal Brief is presented in support of the Notice of Appeal to the Board of Patent Appeals and Interferences, filed on January 26, 2009, from the Final Rejection of claims 1, 5-6 and 12-87 of the above-identified application, as set forth in the Final Office Action mailed on November 12, 2008.

The Commissioner of Patents and Trademarks is hereby authorized to charge Deposit Account No. 19-0743 in the amount of \$540.00 which represents the requisite fee set forth in 37 C.F.R. § 41.20(b)(2). The Appellants respectfully request consideration and reversal of the Examiner's rejections of pending claims.

1. REAL PARTY IN INTEREST

The real party in interest of the above-captioned patent application is the assignee, SAP AG.

2. RELATED APPEALS AND INTERFERENCES

There are no other appeals or interferences known to Appellants that will have a bearing on the Board's decision in the present appeal.

3. STATUS OF THE CLAIMS

The present application was filed on July 17, 2003 with claims 1-77. A preliminary amendment was filed on June 7, 2004 adding claims 78-87. A Non-Final Office Action was mailed on February 27, 2007 rejecting claims 1-87. A response to Non-Final Office Action was filed on June 27, 2007. A Final Office Action was mailed on October 5, 2007 rejecting 1-87. A response to Final Office Action was filed on December 6, 2007, canceling claims 2-4 and 7-11. A Non-Final Office Action was mailed January 10, 2008 rejecting claims 1, 5-6 and 12-87. A response to Non-Final Office Action was filed on April 9, 2008. A Final Office Action was mailed February November 12, 2008 rejecting claims 1, 5-6 and 12-87. A response to Final Office Action was filed on February 10, 2009. Independent claims 1, 20, 33, 40, 59, and 71 stand twice rejected, remain pending, and are the subject of the present Appeal.

4. STATUS OF AMENDMENTS

Amendments have been made subsequent to the Final Office Action dated November 12, 2008. Claims 12, 13, 15, 19, 37-39, 41-58, 60-70, 72-79, and 83-87 are amended. The amendments have been entered according to the Advisory Action mailed February 10, 2009.

5. SUMMARY OF CLAIMED SUBJECT MATTER

Appellants' invention as claimed is directed at a mechanism for collaborative design process. (Title.)

INDEPENDENT CLAIM 1

1. A computer-implemented method comprising:

defining a stored data set maintained by a first entity of a computer system to include a locked data set and an unlocked data set, the stored data set being stored in memory, the unlocked data set being available for modification and the locked data set being protected from modification; [specification, page 2, lines 5-8 and 17-19]

transmitting the locked data set and the unlocked data set to a second entity; [specification, page 3, lines 4-5] and

reversing the locked data set and the unlocked data set at the second entity, such that the locked data set becomes an unlocked data set being available for modification and the unlocked data set becomes a locked data set being protected from modification. [Figure 3; specification, page 9, line 28 - page 10, line 2]

INDEPENDENT CLAIM 20

20. A computer-implemented method of sharing information, comprising:

defining a master data set in a first entity of a computer system, the master data set being stored in memory; [specification, page 2, lines 5-6]

assigning permissions, including permission to change a first subset of data within the master data set based on predetermined criteria, the permissions indicating operations that a second entity may perform on the first subset data and applications that the second entity may use for manipulating the first subset of data; [specification, page 3, lines 9-14]

transmitting a copy of the master data set with indications of the permissions to the second entity, the transmitted copy of the master data set including locked and unlocked data, the locked data in the received copy of the master data set corresponding to unlocked data in the

master data set in the first entity and the unlocked data in the received copy of the master data set corresponding to locked data in the master data set in the first entity; [specification, page 3, lines 11-14] and

receiving a manipulated master data set in accordance with the assigned permissions from the second entity, the manipulated master data set including a second subset of data resulting from the first subset of data being manipulated by the second entity using one or more of the operations indicated in the permissions [specification, page 3, lines 16-18].

INDEPENDENT CLAIM 33

33. A computer-implemented method of sharing information, comprising:

receiving, from a first entity of a computer system, a copy of a master data set, the master data set including locked and unlocked data and being stored in memory, the received copy of the master data set including locked and unlocked data, the locked data in the received copy of the master data set corresponding to the unlocked data in the master data set and the unlocked data in the received copy of the master data set corresponding to the locked data in the master data set; [specification, page 4, lines 10-13]

modifying the copy of the master data set; [specification, page 4, lines 14-15] and
transmitting the modified copy of the master data set to the first entity
[specification, page 4, lines 2-7].

INDEPENDENT CLAIM 40

40. A computer program product, tangibly stored on one or more computer-readable storage devices, the computer program product comprising instructions operable to cause a programmable processor to:

define a stored data set maintained by a first entity to include a locked data set and an unlocked data set, the unlocked data set being available for modification and the locked data set being protected from modification; [specification, page 2, lines 5-8 and 17-19]

transmit the locked data set and the unlocked data set to a second entity
[specification, page 3, lines 4-5]; and

reverse the locked data set and the unlocked data set at the second entity, such that
the locked data set becomes an unlocked data set being available for modification and the
unlocked data set becomes a locked data set being protected from modification. [Figure 3;
specification, page 9, line 28 - page 10, line 2]

INDEPENDENT CLAIM 59

59. A computer program product, tangibly stored on one or more computer-readable storage
devices, the computer program product comprising instructions operable to cause a
programmable processor to:

define a master data set in a first entity; [specification, page 2, lines 5-8]

assign permissions, including permission to change data within the master data set
based on predetermined criteria; [specification, page 3, lines 9-14]

transmit a copy of the master data set with indications of the permissions to the
second entity, the transmitted copy of the master data set including locked and unlocked data, the
locked data in the transmitted copy of the master data set corresponding to unlocked data in the
master data set in the first entity and the unlocked data in the transmitted copy of the master data
set corresponding to locked data in the master data set in the first entity; [specification, page 3,
lines 11-14] and

receive changes to the master data set from the second entity. [specification, page
3, lines 16-18]

INDEPENDENT CLAIM 71

71. A computer program product, tangibly stored on one or more computer-readable storage
devices, the computer program product comprising instructions operable to cause a
programmable processor to:

receive, from a first entity, a copy of a master data set with permissions for using
the master data set, the master data set including locked and unlocked data, the first permissions

allowing changes to the unlocked data and access but no changes to the locked data, the permissions indicating operations that may be performed on the unlocked data and the locked data and applications that the second entity may use for manipulating the unlocked data, the received copy of the master data set including locked and unlocked data, the locked data in the received copy of the master data set corresponding to unlocked data in the master data set in the first entity and the unlocked data in the received copy of the master data set corresponding to the locked data in the master data set in the first entity; [Figure 3; specification, page 9, line 28 - page 10, line 2]

modify the copy of the master data set according to the permissions and user input to generate a modified copy of the master data set, wherein modifying includes applying one or more of the operations indicated in the permissions to the unlocked data; [specification, page 4, lines 14-15] and

transmit the modified copy of the master data set to the first entity. [specification, page 4, lines 2-7]

This summary does not provide an exhaustive or exclusive view of the present subject matter, and Appellant refers to each of the appended claims and its legal equivalents for a complete statement of the invention.

6. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Independent claims 1, 20, 33, 40, 59, and 71 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Carter et al. (U.S. 5,418,945) in view of Hurvig (U.S. 5,628,005).

7. ARGUMENT

As noted above, independent claims 1, 20, 33, 40, 59, and 71 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Carter et al. (U.S. Patent No. 5,418,945) in view of Hurvig (U.S. Patent No. 5,628,005). This rejection is respectfully traversed, Appellants respectfully submit that the Examiner has made an improper *prima facie* showing of obviousness at least because the combination of Carter and Hurvig fails to disclose or suggest all elements of claims 1, 20, 33, 40, 59, and 71.

A) The Applicable Law under 35 U.S.C. §103(a)

Pursuant to 35 U.S.C. §103(a), "[a] patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains." The determination of obviousness under 35 U.S.C. § 103 is a legal conclusion based on factual evidence. *See Princeton Biochemicals, Inc. v. Beckman Coulter, Inc.*, 7, 1336-37 (Fed. Cir. 2005).

Examiner has the burden under 35 U.S.C. § 103 to establish a *prima facie* case of obviousness. *In re Fine*, 837 F.2d 1071, 1074 (Fed. Cir. 1988). To establish *prima facie* obviousness of a claimed invention, all the claim limitations must be taught or suggested by the prior art. M.P.E.P. §2143.03 (citing *In re Royka*, 490 F.2d 981 (CCPA 1974)). "Mere identification in the prior art of each element is insufficient to defeat the patentability of the combined subject matter as a whole." *In re Kahn*, 2006 WL 708687, *9 (Fed. Cir. 2006). Such a teaching or suggestion must be supported by substantial evidence. *Id* at *8. Substantial evidence is something more than a mere scintilla of evidence. *Id*. "Rejections on obviousness grounds cannot be sustained by mere conclusory statements; instead, there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness." *Id* at *10 (quoted in *KSR Int'l Co. v. Teleflex Inc.*, 127 S.Ct. 1727 (2007)).

Applicant can rebut a presumption of obviousness based on a claimed invention that falls within a prior art range by showing "(1) [t]hat the prior art taught away from the claimed invention...or (2) that there are new and unexpected results relative to the prior art." *Iron Grip Barbell Co., Inc. v. USA Sports, Inc.*, 392 F.3d 1317, 1322, 73 USPQ2d 1225, 1228 (Fed. Cir. 2004).

B) Overview of the Technique in Carter

The description provided by Carter relates to file based databases, and to file based databases which are highly available for both read and write access.¹ Carter declares that there exists a need for a hybrid database that combines the features of commercial databases and source code control systems. The hybrid database, according to Carter, must ensure validity and integrity of the files in the database, and yet must be always available for both reading and writing. The hybrid database must be compatible with computer networks, yet must provide a high degree of security for sensitive manufacturing information.² Carter describes a first data control computer system, a second data control computer system, and a third data control computer system to store a first resident database, a second resident database, and a third resident database respectively. The first data control computer system, the second data control computer system, and the third data control computer system are linked together as part of a computer network and each has a respective server. Each of those servers have respective resident databases. Each resident database comprises a hierarchical file structure in which groups of files are stored underneath a common root directory. The groups of files themselves form a similar hierarchy of directories and files within a common file group directory. The database information is thus a plurality of sub-directories within a master database directory, each sub-directory containing a predetermined group of files and sub-directories. The individual files contain the information required for a predetermined design to be fabricated with a predetermined process flow on a predetermined fabrication line. A resident database contains a first master file group, which is a predetermined, hierarchical grouping of files.³ A server comprises a variety of

¹ Carter, 1: 6-9.

² Carter, 2: 43-50.

³ Carter, 3: 17-57.

separable functions: server means for interacting with clients and other servers through a computer network; storing means for fetching file groups from second resident database and for writing new or changed master file groups to second resident database; security means for authorizing or denying access to second resident database; and state means for coordinating the read and write state of the master file groups.⁴ Carter explains, with reference to Figure 4, that when a file group is checked out for alteration, the state means records that file group as being locked. State means ensures that only one client has a copy of the file group checked out for alteration. Were the file group is already in a locked state then, a check out request is denied.⁵

C) Overview of the Technique in Hurvig

The description provided by Hurvig relates to shared resource allocation in a multi-user networking environment, and more particularly, to a system providing opportunistic file access control over shared resources in a multi-user networking environment.⁶ File control features are described in Hurvig with reference to Figure 5. A server in Hurvig waits for a remote process to either request or relinquish access to a file stored at the server resource. Upon receiving an appropriate signal, the server determines whether the remote process is requesting or relinquishing access to a file. If the process has requested access to a file, the server checks its file table to determine whether the requesting process may be given an opportunistic lock on the requested file. An opportunistic lock is granted if 1) the requesting process is the only process requesting access to the file, or 2) the requesting process is seeking read-only access to the file and no other processes have read-write access to the file. If the opportunistic lock is granted, then the server transmits the file to the process using the first socket of the requesting process, updates its file table to indicate the requesting process has access to the file and the type of access (e.g., read-only or read-write); and notifies the requesting process that it may maintain the file locally, in its local cache or other storage resource available to it. If an opportunistic lock is not granted, then the server notifies the remote process that the file must be maintained at the server.⁷

⁴ Carter, 4: 37-47.

⁵ Carter, 5: 18-26.

⁶ Hurvig, 1: 8-12.

⁷ Hurvig, 6-27.

D) The Rejection of Independent claims 1, 20, 33, 40, 59, and 71 under 35 U.S.C. §.103(a) over Carter in view of Hurvig is Improper

The Examiner has the burden under 35 U.S.C. § 103 to establish a *prima facie* case of obviousness.⁸ To establish *prima facie* obviousness of a claimed invention, all claim limitations must be taught or suggested by the prior art.⁹ Thus, unless it is shown that the proposed combination of the cited documents relied upon by the Examiner for obviousness rejections, namely Carter and Hurvig, discloses or suggests all elements of a claim, the claim is patentable in view of Carter and Hurvig.

Carter is directed at providing a hybrid database that combines the features of commercial databases and source code control systems.¹⁰ The Office action cites the resident database that contains the first master file group¹¹ to show the stored data set recited in claim 1. Carter explains that when a file group is checked out by a client for alteration, the server records the master file group as locked. There is no indication that a file group transferred to the client in response to the client's request is a locked file group. On the contrary, Carter states that the file group is transferred to the client for the purposes of alteration, which suggests that the server, transfers an unlocked copy of the requested file group to the client. When, on the other hand, a master file group is locked on the server, a check out request from a client is denied.¹² Carter, therefore, does not disclose and also teaches away from "transmitting the locked data set... to a second entity" recited in claim 1.

The feature of "transmitting the locked data set... to a second entity" recited in claim 1 is not present in Carter (as explained above) and also is not present in Hurvig that does not contemplate maintaining a locked data set at one entity and then transmitting the locked data set to another entity. There is no conceivable combination of Carter (that teaches away from "transmitting the locked data set... to a second entity") and Hurvig (that does not contemplate transmitting a locked data set to another entity) that would yield "transmitting the locked data

⁸ *In re Fine*, 837 F.2d 1071, 1074 (Fed. Cir. 1988).

⁹ *In re Royka*, 490 F.2d 981 (CCPA 1974).

¹⁰ Carter, 2: 48-50.

¹¹ Carter, 3: 52-57

¹² Carter, 5: 18-26.

set... to a second entity" recited in claim 1. Thus, the feature of "transmitting the locked data set... to a second entity" recited in claim 1 is not disclosed or suggested by the combination of Carter and Hurvig.

In the Advisory Action, the Examiner asserts that Carter discloses "reversing the locked data set ... such that the locked data set becomes an unlocked data set being available for modification" and refers, again, to the passage in Carter at 5: 18-26 (discussed immediately above). As explained above, in Carter, a check out request from a client is denied if a master file group is locked on the server and therefore, because "transmitting the locked data set... to a second entity" never takes place in Carter, the operation of reversing thus transmitted locked data set at the second entity is meaningless in the context of Carter. Combining Carter with Hurvig (that does not refer to files stored in the resource 208 as being "locked" or "unlocked" but instead discusses the concept of an opportunistic lock that may be selectively granted to a requesting process¹³) does not remedy this deficiency of Carter and does not yield "reversing the locked data set ... such that the locked data set becomes an unlocked data set being available for modification" recited in claim 1.

Thus, the combination of Carter and Hurvig fails to disclose or suggest a method comprising "reversing the locked data set and the unlocked data set at the second entity, such that the locked data set becomes an unlocked data set being available for modification and the unlocked data set becomes a locked data set being protected from modification," as recited in claim 1. Because the combination of Carter and Hurvig fails to disclose or suggest all elements of claim 1, claim 1 is patentable and should be allowed.

Furthermore, combining Hurvig with Carter does not does not yield the method of claim 1, because combining Carter (transferring a copy of a file group to a client and then preventing access to the corresponding maser file group by other clients) and Hurvig (granting a process an opportunistic lock on a file to prevent other processes from obtaining a copy of the file) can never result in a scenario where a locked data set and an unlocked data set are maintained at a first entity as a single data set, the unlocked data set is available for modification and the locked

¹³ Hurvig, 9: 14-24.

data set is protected from modification, but at a second entity that unlocked data set becomes locked and that locked data set becomes unlocked. The Examiner suggests, in the Advisory Action, that unlocked and locked data described in the references yields the inventions of our independent claims because "simply combining the two data sets into one was a known technique." The Examiner does not, however, take into consideration the fact that the "locked" and "unlocked" states of a data set are mutually exclusive in the prior art, such as in Carter, where a file group is either available for checking out when unlocked or not available for checking out when it is locked.¹⁴

Thus, the combination of Carter and Hurvig fails to disclose or suggest the features recited in claim 1. Because the combination of Carter and Hurvig fails to disclose or suggest all elements of claim 1, claim 1 is patentable and should be allowed. It is respectfully requested that the rejection of claim 1 be reversed.

The reasoning with articulated above is applicable to independent claims 20, 33, 40, 59, and 71 *mutatis mutandis*. Claims 20, 33, 40, 59, and 71 are therefore patentable for the reasons articulated above. It is respectfully requested that the rejection of claims 1, 20, 33, 40, 59, and 71 be reversed.

¹⁴ Carter, 5: 8-26.

SUMMARY

The reasons argued above are summarized as follows. First, Carter teaches away from the invention of claims 1, 20, 33, 40, 59, and 71. Second, the combination of Carter and Hurvig fails to disclose or suggest every element recited in independent claims 1, 20, 33, 40, 59, and 71. For the reasons articulated above, with respect to claims 1, 20, 33, 40, 59, and 71, the Examiner failed to make *prima facie* showing of obviousness under 35 USC § 103(a) in view of the combination of Carter and Hurvig. It is respectfully submitted that the art cited does not render claims 1, 20, 33, 40, 59, and 71 obvious and that the claims are patentable over the cited art. Reversal of the rejection and allowance of the pending claims are respectfully requested.

Respectfully submitted,

SCHWEGMAN, LUNDBERG & WOESSNER, P.A.
P.O. Box 2938
Minneapolis, MN 55402

Date March 26, 2009 By /Elena Dreszer/
Elena B. Dreszer
Reg. No. 55,128

CERTIFICATE UNDER 37 CFR 1.8: The undersigned hereby certifies that this correspondence is being filed using the USPTO's electronic filing system EFS-Web, and is addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on this 26 day of March 2009.

Dawn R. Shaw Dawn R. Shaw
Name Signature

8. CLAIMS APPENDIX

1. A computer-implemented method comprising;

defining a stored data set maintained by a first entity of a computer system to include a locked data set and an unlocked data set, the stored data set being stored in memory, the unlocked data set being available for modification and the locked data set being protected from modification;

transmitting the locked data set and the unlocked data set to a second entity; and

reversing the locked data set and the unlocked data set at the second entity, such that the locked data set becomes an unlocked data set being available for modification and the unlocked data set becomes a locked data set being protected from modification.

5. The method of claim 1, wherein the defining of the stored data set maintained by the first entity to include the locked data set and the unlocked data set includes defining the locked data set to include information to call an application and the unlocked data set to include data to be used by the application.

6. The method of claim 1, wherein the defining of the stored data set maintained by the first entity to include the locked data set and the unlocked set includes defining version data for an application as the locked data set and defining raw data for the second entity to look at or use as the unlocked data.

12. The method of claim 1, wherein defining the stored data set to include the locked data set and the unlocked data set further includes defining the stored data set to include a restricted data set including data that is not part of the locked data set or the unlocked data set.

13. The method of claim 1, wherein defining the stored data set to include the locked data set and the unlocked data set further includes assigning data to the locked data set based on closeness criteria.

14. The method of claim 13, wherein assigning data to the locked data set based on closeness criteria includes assigning data to the locked data set based on at least one of geometric closeness, organizational closeness, and collective closeness.

15. The method of claim 1, wherein defining the stored data set to include the locked data set and the unlocked data set includes assigning data in the stored data set to the locked data set and the unlocked data set based on a function of the second entity.

16. The method of claim 1, wherein defining the stored data set to include a locked data set and an unlocked data set includes defining the stored data set to include a locked data set and an unlocked data set for the second entity, the method further comprising:

defining data included in the unlocked data set for the second entity as locked for other entities.

17. The method of claim 16, wherein defining data included in the unlocked data set for the second entity as locked for other entities includes defining data included in the unlocked data set for the second entity as locked for all other entities during a period of time when the second entity has access to the unlocked data set.

18. The method of claim 1, further comprising:

receiving modified data from the second entity; and
integrating the modified data into the stored data set.

19. The method of claim 1, wherein defining the stored data set to include the locked data set and the unlocked data set includes defining the stored data set to include a locked data set and an unlocked data set based on user input.

20. A computer-implemented method of sharing information, comprising:

defining a master data set in a first entity of a computer system, the master data set being stored in memory;

assigning permissions, including permission to change a first subset of data within the master data set based on predetermined criteria, the permissions indicating operations that a second entity may perform on the first subset data and applications that the second entity may use for manipulating the first subset of data;

transmitting a copy of the master data set with indications of the permissions to the second entity, the transmitted copy of the master data set including locked and unlocked data, the locked data in the received copy of the master data set corresponding to unlocked data in the master data set in the first entity and the unlocked data in the received copy of the master data set corresponding to locked data in the master data set in the first entity; and

receiving a manipulated master data set in accordance with the assigned permissions from the second entity, the manipulated master data set including a second subset of data resulting from the first subset of data being manipulated by the second entity using one or more of the operations indicated in the permissions.

21. The method of claim 20, further comprising receiving a modified copy of the master data set from the second entity and integrating the modified copy of the master data set with the master data set.

22. The method of claim 21, wherein receiving the modified copy of the master data set includes receiving additional data.

23. The method of claim 21, wherein receiving the modified copy of the master data set includes receiving changed data.

24. The method of claim 23, wherein receiving changed data includes receiving data that has been changed in response to design considerations.

25. The method of claim 20, wherein assigning permissions includes assigning authority to read data.

26. The method of claim 20, wherein assigning permissions includes assigning authority to change data that is a subset of the transmitted copy of the master data.

27. The method of claim 20, wherein assigning permissions includes assigning authority to add data.

28. The method of claim 20, wherein assigning permissions includes assigning authority to delete data.

29. The method of claim 20, wherein assigning permissions includes assigning authority to access predetermined types of data within the subset.

30. The method of claim 20, wherein assigning permissions includes assigning permissions based on at least one of an identity of an entity, a function of the entity and a user's position within the entity.

31. The method of claim 30, wherein assigning permissions based on the user's position within the entity includes assigning permissions according to a hierarchy within the second entity so that a highest ranking member of an entity has a greater number of permissions, and a number and extent of permissions decrease as rank decreases.

32. The method of claim 20, wherein assigning permissions includes assigning different permissions for different subsets of the unlocked data.

33. A computer-implemented method of sharing information, comprising:

receiving, from a first entity of a computer system, a copy of a master data set, the master data set including locked and unlocked data and being stored in memory, the received

copy of the master data set including locked and unlocked data, the locked data in the received copy of the master data set corresponding to the unlocked data in the master data set and the unlocked data in the received copy of the master data set corresponding to the locked data in the master data set;

modifying the copy of the master data set; and

transmitting the modified copy of the master data set to the first entity.

34. The method of claim 33, wherein receiving the copy of the master data set in a second entity includes receiving the copy of the master data set in a computer application.

35. The method of claim 34, wherein receiving the copy of the master data set in a computer application includes receiving version information regarding the computer application in the locked data and receiving raw data for manipulation in the unlocked data.

36. The method of claim 33, wherein modifying the copy of the master data set includes performing design processes on the unlocked portion of the data.

37. The method of claim 33, wherein receiving the copy of the master data set includes receiving permissions to do at least one of read, change, delete and add data to the unlocked data.

38. The method of claim 33, wherein receiving the copy of the master data set includes receiving the copy of the master data set with permissions based on subsets of the unlocked data, with different permissions assigned for different subsets of the unlocked data.

39. The method of claim 33, wherein receiving the copy of the master data set includes receiving the copy of the master data set with permissions based on at least one of an identity of the second entity, a function of the second entity and a hierarchy of users within the second entity.

40. A computer program product, tangibly stored on one or more computer-readable storage devices, the computer program product comprising instructions operable to cause a programmable processor to:

define a stored data set maintained by a first entity to include a locked data set and an unlocked data set, the unlocked data set being available for modification and the locked data set being protected from modification

transmit the locked data set and the unlocked data set to a second entity; and

reverse the locked data set and the unlocked data set at the second entity, such that the locked data set becomes an unlocked data set being available for modification and the unlocked data set becomes a locked data set being protected from modification.

41. The computer program product of claim 40, wherein the instructions include instructions operable to cause the programmable processor to provide an application in a computer system with access to the stored data set.

42. The computer program product of claim 41, wherein the instructions include instructions operable to cause the programmable processor to provide an application maintained at a location external to the first entity with access to the stored data set.

43. The computer program product of claim 41, wherein the instructions include instructions operable to cause the programmable processor to provide a computer aided design system with access to the stored data set.

44. The computer program product of claim 41, wherein the instructions include instructions operable to cause the programmable processor to define the locked data set to include information to call an application and to define the unlocked data set to include data to be used by the application.

45. The computer program product of claim 41, wherein the instructions include instructions operable to cause a the programmable processor to define as the locked data set version data for

the application and to define as the unlocked data set raw data for the second entity to look at or use.

46. The computer program product of claim 40, wherein the instructions include instructions operable to cause the programmable processor to send the stored data set to the second entity.

47. The computer program product of claim 40, further comprising instructions operable to cause the programmable processor to provide the first entity with access to the stored data set, the first entity having permission to view the unlocked data set and to change only the locked data set.

48. The computer program product of claim 40, wherein the instructions include instructions operable to cause the programmable processor to provide a computer aided design system with access to the stored data set.

49. The computer program product of claim 40, wherein the instructions include instructions operable to cause the programmable processor to provide an entity that is external to the first entity with access to the stored data set.

50. The computer program product of claim 40, wherein the instructions include instructions operable to cause the programmable processor to assign, based on predetermined criteria, data in the stored data set to a locked data set and an unlocked data set.

51. The computer program product of claim 40, wherein the instructions include instructions operable to cause the programmable processor to include a restricted data set including data that is not part of the locked data set or the unlocked data set.

52. The computer program product of claim 40, wherein the instructions include instructions operable to cause the programmable processor to assign data to the locked data set based on a closeness criteria.

53. The computer program product of claim 52, wherein the instructions include instructions operable to cause the programmable processor to assign data to the locked data set based on at least one of geometric closeness, organizational closeness, and collective closeness.

54. The computer program product of claim 40, wherein the instructions include instructions operable to cause the programmable processor to assign data in the stored data set to the locked data set and the unlocked data set based on a function of the second entity.

55. The computer program product of claim 40, wherein the instructions include instructions operable to cause the programmable processor to define the stored data set to include the locked data set and the unlocked data set for the second entity, the computer program product further comprising instructions operable to cause the programmable processor to:

define data included in the unlocked data set for the second entity as locked for all other entities.

56. The computer program product of claim 55, wherein the instructions operable to cause the programmable processor to define data included in the unlocked data set for the second entity as locked for all other entities include instructions operable to cause the programmable processor to define data included in the unlocked data set for the second entity as locked for other entities during a period of time when the second entity has access to the unlocked data set.

57. The computer program product of claim 40, further comprising instructions operable to cause the programmable processor to:

transmit data from the stored data set to the second entity;

receive modified data from the second entity; and

integrate the modified data corresponding to the unlocked data set into the stored data set.

58. The computer program product of claim 40, wherein the instructions include instructions operable to cause the programmable processor to define, based on user input, the stored data set to include the locked data set and the unlocked data set.

59. A computer program product, tangibly stored on one or more computer-readable storage devices, the computer program product comprising instructions operable to cause a programmable processor to:

define a master data set in a first entity;

assign permissions, including permission to change data within the master data set based on predetermined criteria;

transmit a copy of the master data set with indications of the permissions to the second entity, the transmitted copy of the master data set including locked and unlocked data, the locked data in the transmitted copy of the master data set corresponding to unlocked data in the master data set in the first entity and the unlocked data in the transmitted copy of the master data set corresponding to locked data in the master data set in the first entity; and

receive changes to the master data set from the second entity.

60. The computer program product of claim 59, further comprising instructions operable to cause the programmable processor to receive a modified copy of the master data set from the second entity and to integrate the modified copy of the master data set with the master data set.

61. The computer program product of claim 60, wherein the instructions operable to cause the programmable processor to receive the modified copy of the master data set include instructions operable to cause a programmable processor to receive additional data.

62. The computer program product of claim 60, wherein the instructions operable to cause the programmable processor to receive the modified copy of the master data set include instructions operable to cause a programmable processor to receive changed data.

63. The computer program product of claim 62, wherein the instructions operable to cause the programmable processor to receive changed data include instructions operable to cause a programmable processor to receive data that has been changed in response to design considerations.

64. The computer program product of claim 59, wherein the instructions operable to cause the programmable processor to assign permissions include instructions operable to cause a programmable processor to assign authority to read data.

65. The computer program product of claim 59, wherein the instructions operable to cause the programmable processor to assign permissions include instructions operable to cause a programmable processor to assign authority to change data that is a subset of the transmitted copy of the master data.

66. The computer program product of claim 59, wherein the instructions operable to cause the programmable processor to assign permissions include instructions operable to cause a programmable processor to assign authority to add data.

67. The computer program product of claim 59, wherein the instructions operable to cause the programmable processor to assign permissions include instructions operable to cause a programmable processor to assign authority to delete data.

68. The computer program product of claim 59, wherein the instructions operable to cause the programmable processor to assign permissions include instructions operable to cause a programmable processor to assign authority to access predetermined types of data within the subset.

69. The computer program product of claim 59, wherein the instructions operable to cause the programmable processor to assign permissions include instructions operable to cause a

programmable processor to assign permissions based on at least one of an identity of an entity, a function of the entity and a user's position within the entity.

70. The computer program product of claim 69, wherein the instructions operable to cause the programmable processor to assign permissions based on a user's position within the entity include instructions operable to cause a programmable processor to assign permissions according to a hierarchy within a department so that a highest ranking member of a department has a greater number of permissions, and a number and extent of permissions decrease as rank decreases.

71. A computer program product, tangibly stored on one or more computer-readable storage devices, the computer program product comprising instructions operable to cause a programmable processor to:

receive, from a first entity, a copy of a master data set with permissions for using the master data set, the master data set including locked and unlocked data, the first permissions allowing changes to the unlocked data and access but no changes to the locked data, the permissions indicating operations that may be performed on the unlocked data and the locked data and applications that the second entity may use for manipulating the unlocked data, the received copy of the master data set including locked and unlocked data, the locked data in the received copy of the master data set corresponding to unlocked data in the master data set in the first entity and the unlocked data in the received copy of the master data set corresponding to the locked data in the master data set in the first entity;

modify the copy of the master data set according to the permissions and user input to generate a modified copy of the master data set, wherein modifying includes applying one or more of the operations indicated in the permissions to the unlocked data; and

transmit the modified copy of the master data set to the first entity.

72. The computer program product of claim 71, wherein the instructions include instructions operable to cause the programmable processor include instructions operable to cause the programmable processor to receive the copy of the master data set in a computer application.

73. The computer program product of claim 72, wherein the instructions include instructions operable to cause the programmable processor to receive, in the locked data, version information regarding the computer application and to receive, in the unlocked data, raw data for manipulation.

74. The computer program product of claim 72, wherein the instructions operable to cause the programmable processor to modify the copy of the master data set include instructions operable to cause the programmable processor to perform design processes on the unlocked portion of the data.

75. The computer program product of claim 74, wherein the instructions operable to cause the programmable processor to receive the copy of the master data set with permissions for using the master data set include instructions operable to cause the programmable processor to receive permissions to do at least one of read, change, delete and add data to the unlocked data.

76. The computer program product of claim 71, wherein the instructions operable to cause the programmable processor to receive the copy of the master data set with permissions for using the master data set include instructions operable to cause the programmable processor to receive the copy of the master data set with permissions based on subsets of the unlocked data, with different permissions assigned for different subsets of the unlocked data.

77. The computer program product of claim 71, wherein the instructions operable to cause the programmable processor to receive the copy of the master data set with permissions for using the master data set include instructions operable to cause the programmable processor to receive the copy of the master data set with permissions based on at least one of an identity of the second entity, a function of the second entity and a hierarchy of users within the second entity.

78. The method of claim 1 including providing a testing application with access to the stored data set.

79. The method of claim 1, including providing a testing entity with access to the stored data set.

80. The method of claim 22, wherein receiving additional data includes receiving test results.

81. The method of claim 23, wherein receiving changed data includes receiving data that has been changed in response to testing.

82. The method of claim 33, wherein modifying the copy of the master data set includes performing testing on the unlocked portion of the data.

83. The computer program product of claim 41, wherein the instructions operable to cause the programmable processor to provide the application in the computer system with access to the stored data set includes the instructions operable to cause the programmable processor to provide a testing application with access to the stored data set.

84. The computer program product of claim 40, wherein the instructions operable to cause the programmable processor to provide the second entity with access to the stored data set includes the instructions operable to cause the programmable processor to provide a testing entity with access to the stored data set.

85. The computer program product of claim 61, wherein the instructions operable to cause the programmable processor to receive the additional data include instructions operable to cause the programmable processor to receive test results.

86. The computer program product of claim 62, wherein the instructions operable to cause the programmable processor to receive the changed data include instructions operable to cause the programmable processor to receive data that has been changed in response to testing.

87. The computer program product of claim 72, wherein the instructions operable to cause the programmable processor to modify the copy of the master data set include instructions operable to cause the programmable processor to perform testing on the unlocked portion of the data.

9. EVIDENCE APPENDIX

None.

10. RELATED PROCEEDINGS APPENDIX

None.